APPLICATION

FOR

UNITED STATES LETTERS PATENT

APPLICANT NAME: J. P. Goddard

TITLE: SYSTEM, METHOD AND PROGRAM PRODUCT TO DETERMINE SECURITY RISK OF AN APPLICATION

DOCKET NO.: END920030107US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

Certificate of Mailing Under 37 CFR 1.10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 as "Express Mail Post Office to Addressee"

"Express Mail" Label No.: EV 342658922US

On: 10/21/2003

Denise M. Jurik

Typed or Printed Name of Person Mailing Correspondence

Denise M Jurik 10/21/03

Signature **Date**

SYSTEM, METHOD AND PROGRAM PRODUCT TO DETERMINE SECURITY RISK OF AN APPLICATION

Background of the Invention

The invention relates generally to computer systems, and deals more particularly with a technique to determine a security risk of a software application.

Every software application poses some security risk. The risks include unauthorized access, attack by hackers, computer viruses and worms, loss or corruption of data, loss of availability to data or the application and theft of proprietary or personal data. The vulnerabilities can be caused by programming errors, configuration problems or application design errors. Often, IT organizations will undergo a certification process to ensure that an application being considered for deployment meets some minimum standards for IT security. Known certification processes comprise some form of design review, security technical testing, and risk analysis based on the results of the design review and testing. The known certification process may also balance security risks with business needs, and to some extent is subjective.

There is an existing standard, NIST 800-37, for certifying applications used for the US government. This standard comprises guidelines for certifying and accrediting information systems that support the executive agencies of the U.S. Federal government.

An object of the present invention is to provide a system, method and program product for determining if a software application is sufficiently secure, under the circumstances and considering the business needs, to be deployed.

Another object of the present invention is to provide a system, method and program product of the foregoing type which is more objective than other known certification processes.

Summary of the Invention

The invention resides in a system, method and program product for evaluating a security risk of an application. A determination is made whether unauthorized access or loss of data maintained or accessed by the application would cause substantial damage. A determination is made whether the application is shared by different customers. A determination is made whether a vulnerability in the application can be exploited by a person or program which has not been authenticated to the application or a system in which the application runs. A numerical value or weight is assigned to each of the foregoing determinations. Each of the numerical values or weights corresponds to a significance of the determination in evaluating said security risk. The numerical values or weights are combined to evaluate the security risk.

The invention also resides in another system, method and program product for a security risk of an application. A determination is made whether unauthorized access or loss of the data would cause substantial damage. A determination is made whether the application is vulnerable to attack by a third party. A determination is made whether the application is shared by different customers. A determination is made as to mitigation controls for the security risk of the application. A numerical value or weight is assigned to each of the foregoing determinations. Each of the numerical values or weights corresponds to a significance of the determination in evaluating the security risk.

Brief Description of the Figures

Figure 1 is a flow chart illustrating a process according to the present invention for determining sensitivity of data maintained or accessed by an application being considered for deployment.

Figure 2 is a flow chart illustrating a process according to the present invention for determining an extent of vulnerability caused by the application of Figure 1.

Figure 3 is a flow chart illustrating a process according to the present invention for determining a sensitivity of a customer of the application of Figure 1.

Figure 4 is a flow chart illustrating a process according to the present invention for determining the level of assurance provided by mitigation controls for the application of Figure 1.

Detailed Description of the Preferred Embodiments

The present invention comprises several processes for evaluating different factors of and relating to an application which is being considered for deployment. These factors are the sensitivity of the data maintained or accessed by the application, the vulnerability caused by the application, the sensitivity of the customer of the application, and mitigation controls for the risk caused by the application. Each of the processes yields a numerical score. All the numerical scores are added together to form a combined score representing a security risk posed by the application in the environment and under the circumstances in which the application will be deployed. Then, the combined score is compared to the business needs and business value to determine whether the application should be deployed.

Figure 1 illustrates a process 100 for evaluating sensitivity of data maintained or accessed by the application. The scope of this data is defined as that which resides on a same logical or virtual system where the application resides, or that data which the application is intended or expected to access. For example, if an application resides on a Linux host with IP address 192.168.1.1, and the application allows a user to query the user's account information on a data base manager back end which resides on IP address 192.168.1.1, then all data at IP address 192.168.1.1 is in scope. Also, all data managed by the database manager at IP address 192.168.1.1 is in scope. If there were other virtual machines at IP address 192.168.1.1, then only the data of the virtual host running the application would be in scope. The sensitivity of the data maintained or accessed by the application is that of the most sensitive data in the data scope. In the illustrated embodiment of the present invention, there are three levels of sensitivity - "confidential", "personal" and "aggregation". Before describing the steps of Figure 1, these terms are defined as follows:

"Confidential" data is that which is proprietary to any customer, and if accessed by another party, deleted or corrupted, could result in damage to either the service provider or the customer. "Damage" includes loss of revenue, loss of market share, lawsuits by the customer's

customers, loss of reputation or damage to investors. The data is considered "confidential" as long as the potential damage is significant.

"Personal" data is that which is personal to individuals such as social security numbers, credit cart numbers, bank account numbers, medical information and other information an individual has not put in the public domain. If personal data is accessed by another party, deleted or corrupted, there could be lawsuits from the individuals.

"Aggregation" data is that which could lead to confidential or personal data. In other words, if aggregation data is accessed by another party, the other party could derive information from the aggregation data that is confidential. For example, the other party might be able to piece together various problem tickets to learn that the service provider is about to down size. As another example, the other party might be able to piece together a collection of public documents about another company to learn that the customer is about to acquire this other company.

Figure 1 is a flow chart of a manual process, or an automated process 100 performed by a computer program (with user input of the values), to determine the sensitivity value for the data. In step 101, the process 100 is invoked or begun. An initial value of "one" is set for the sensitivity value to reflect a minimum sensitivity and value of all data (step 102). Then, a determination is made whether any of the data in the data scope is confidential (decision 104). If so, the sensitivity value is increased by "two" (step 106). Then, a determination is made whether a theft, deletion or corruption of the data might cause a substantial damage (decision 108). A substantial damage can be defined as enough liability to warrant public disclosure, enough to affect overall corporation profitability or above 0.5% of business unit base revenue. If so, the sensitivity value is increase by "five" (step 110). After decision 104, no branch, decision 108, no branch or step 110, a determination is made whether any of the data in the data scope is personal in nature (decision 112). If so, the sensitivity value is increased by "one" (step 114). After decision 112, no branch or step 114, a determination is made whether any of the data in the data scope is subject to aggregation (decision 116). If so, the sensitivity value is increased by "one". After decision 118, no branch or step 118, the final sensitivity value results (step 120).

Figure 2 illustrates a process 200 for evaluating an extent of vulnerability caused by the application considered for deployment. Process 200 can be performed manually, or automatically by a computer program with user input. A "vulnerability" is a condition or property of the application that can lead to (a) third party access to data, (b) deletion or corruption of data, (c) administrative access to the application or system in which the application runs or is intended or expected to access, (d) shut down of the application, (e) shut down of the system in which the application is being run or which the application is intended or expected to access, authenticate, manage or monitor. Examples of vulnerabilities are as follows: buffer overflow, SQL injection and race conditions. The present invention considers not only the existence of a vulnerability, but the ease of exploitation of the vulnerability and the potential damage caused by the exploitation. The extent of vulnerability is limited to the system in which the application is run or the systems and data which the application is intended or expected to access.

In step 201, the process 200 is invoked or begun and in step 202, a vulnerability score is initialized to zero. Then, a determination is made whether there are any vulnerabilities in or created by the application considered for deployment (step 203). This determination is based on a security review of the application. The security review comprises a scan of all known error conditions as well as manual checks to attempt to manipulate programming and design errors in the application. If no vulnerabilities were found, then process 200 ends with the vulnerability score being "zero" (step 204). However, if a vulnerability was found (decision 202, yes branch), then a determination is made whether an exploit is currently known to exist or could easily be created (decision 206). Examples of types of known exploits are CERT Advisory CA-2001-19 "Code Red" CERT® Advisory CA-2003-12 Buffer Overflow in Sendmail and telnet vulnerabilities due to passwords being sent in clear text. An example of an exploit that does not currently exist but could easily be created is as follows. If the application will deny service when it is flooded with data packets, then there would be a nonexistent but easy to create exploit because a hacker could simply flood the application with data packets. If there is such an existing exploit or easy to create exploit, then a value of "two" is added to the vulnerability score

(step 210). After decision 206, no branch or step 210, a determination is made as to what the vulnerability will allow (decision 212). The illustrated embodiment of process 200 divides the nature of the vulnerability into five categories, although it is possible to define other categories as well. If the vulnerability allows denial of service of the application, the system in which the application runs, or the system in which the application is intended or expected to access (decision 213), then the vulnerability score is increased by "one" (step 214). If the vulnerability allows read access to any data (decision 217), then the vulnerability score is increased by "one" (step 218). If the vulnerability allows write access to any data (decision 221), then the vulnerability score is increased by "two" (step 222). If the vulnerability allows a third party to have administrative authority to any data maintained or accessed by the application (decision 225), then the vulnerability score is increased by "three" (step 226). If the vulnerability has any other significant effect (decision 227), then the vulnerability score is increased by "one" (step 228). It is possible that a single vulnerability can encompass more than one of the foregoing categories. So, the highest value is chosen from those resulting from decisions 213, 217, 221, 225 or 227 to increase the vulnerability score (step 230). Next, a determination is made whether the vulnerability can be exploited by a person or program which has not been authenticated to the application or system in which the application runs (decision 240). This would increases the risk dramatically, so in such a case, the vulnerability score is doubled (step 242).

Figure 3 illustrates a process 300 for evaluating an extent of sensitivity of the customer which uses the application or which the application serves. The sensitivity considers whether the application is accessible to the customer, or is hidden from the customer. For example, an application used by desk side support personnel of a service provider would have a lower sensitivity than an application used directly by a customer such as a website to submit problems to a service desk. The reason is that a failure of this application would cause a lesser burden on the customer and would have fewer people accessing it thereby lowering the probability of a malicious user accessing the application. As another example, an application used by a systems administrator to automate administration of a system would be hidden from the customer, and therefore have a lower sensitivity than an application used directly by a customer.

An application can be shared by more than one customer. The "sensitivity" also considers how many customers are using the application. An application used by more than one customer would have a higher sensitivity than an application used by only one customer, because it is more likely that one customer could improperly access the other customer's data (despite segregation of data and other security measures) or disrupt or overburden the application so that it cannot support the other customers.

The "sensitivity" also considers how important is the customer and whether the customer is subject to industrial or other external compliance controls or acting as a fiduciary for others. (There are often compliance controls in the financial or medical industry.) The importance of a customer can be measured by the dollar amount of business in absolute terms or in comparison to that of other customers, contracts or business partnerships. Officers of corporations, public accounting firms and investment firms are examples of people who act as fiduciaries for others.

In step 301, the process 300 is invoked or begun and in step 302, a customer sensitivity score is initialized to zero. Then, a determination is made whether at least one customer has direct access to the application (step 303). If so, a value of "one" is added to the score (step 306). Then (or if no customer has direct access to the application), a determination is made whether the application is shared by two or more customers (decision 308). If so, a value of "six" is added to the score (step 310). Then (or if the application is not shared), a determination is made whether any of the customers or users of the application have high importance (decision 314). If so, a value of "three" is added to the score (step 316), and this completes the scoring for customer sensitivity for this scenario (step 320). Referring again to decision 314, no branch, if none of the customers or users of the application have high importance, then a determination is made whether the application is subject to industrial or other external compliance controls (decision 330). If so, than a value of "three" is added to the score (step 332), and this completes the scoring for customer sensitivity in these scenarios (step 320).

Figure 4 illustrates a process 400 for evaluating an extent of controls to reduce or mitigate vulnerability or its effect. The mitigation controls comprise intrusion detection, vulnerability scanning, health checking, network isolation and logging of exploits. An "intrusion detection" system searches each incoming message packet for known signatures of viruses and worms. Intrusion detection also comprises detection of known IT security attacks, alerting service providers of such attacks and potential blocking mechanisms to stop such attacks. Some intrusion detection systems are geared for an entire computer system and others for a specific application. An intrusion detection system geared for a specific application is usually more effective for the application than the more general intrusion detection system for the system because it will have specific attack signatures for that application whereas general intrusion detection systems may not. "Vulnerability scanning" comprises a series of procedures performed on a system or an application to determine if the system or application is susceptible to known vulnerabilities. These procedures include port discovery, enumeration and testing and reporting of discovered vulnerabilities. These procedures can be performed manually or automatically. Generally, the automatic procedure is more comprehensive and less prone to errors than the manual process. Generally, a vulnerability scan for a specific application is more effective for the application than the more general vulnerability scan for the system. "Health checking" comprises an assessment of whether a system or application complies with a specific security policy. This assessment comprises configuration discovery, comparison of configuration to policy and reporting of policy violations to the service provider. These procedures can be performed manually or automatically. Generally, the automatic procedure is more comprehensive and less prone to errors. Generally, a health check for a specific application is more effective for the application than the more general health check for the system. "Strong" network isolation means that the application's host is physically isolated or "segmented" from other hosts (to prevent communication between the two hosts) with strong authentication required to connect to other hosts. "Strong" authentication may require a userID and password or something equivalent. "Weak" authentication may be based on network flow control rules such as source IP address only. This will decrease the chance that a problem with an application on one host will propagate to other hosts. Lesser network isolation occurs when the application is separated from other vital hosts based on effective network flow controls or rules. "Logging"

means that the application logs all malicious activity which is detected or suspected. This log can be used for subsequent intrusion analysis, and if an intrusion is confirmed, to create and install a safeguard for a subsequent intrusion of this type.

In step 401, the process 400 is invoked or begun and in step 402, a mitigation control score is initialized to zero. Then, a determination is made whether a high quality, round the clock, intrusion detection system ("IDS") is in use for the application or the system on which the application runs (decision 404). If so, a value of "three" is subtracted from the initial mitigation control score (step 408). Then, a determination is made whether the intrusion detection system is specific to the application (decision 410). If so, another value of "three" is subtracted from the mitigation control score (step 412). After decision 404, no branch, decision 410, no branch, or step 412, a determination is made whether high quality, automated, periodic vulnerability scanning is performed for the application or the system on which the application runs (decision 420). If so, a value of "two" is subtracted from the mitigation control score (step 422). Then, a determination is made whether the vulnerability scanning is specific to the application (decision 424). If so, another value of "two" is subtracted from the mitigation control score (step 426). After decision 420, no branch, decision 424, no branch, or step 426, a determination is made whether high quality, automated, periodic health checking is performed for the application or the system on which the application runs (decision 430). If so, a value of "two" is subtracted from the mitigation control score (step 422). Then, a determination is made whether the health checking is specific to the application (decision 434). If so, another value of "two" is subtracted from the mitigation control score (step 436). After decision 430, no branch or decision 434, no branch, or step 436, a determination is made whether there is "strong" network isolation (decision 440). If so, a value of "four" is subtracted from the mitigation control score (step 444). After decision 440, no branch or step 444, a determination is made whether there is some but not "strong" network isolation (decision 450). If so, a value of "one" is subtracted from the mitigation control score (step 454). After decision 450, no branch or step 454, a determination is made whether the application logs malicious activity which is detected or suspected (decision 460). If so, a value of "one" is subtracted from the mitigation control score (step 464). Then, all the scores generated by processes 100, 200, 300 and 400 are added together to arrive at the final

score for the application (step 500). (The foregoing are just examples of scores that can be used for the various factors; other scores can be used as well.)

The final score can be used as described below to determine the risk/benefit associated with the application. First, in the illustrated example, the score is classified into a risk category as follows. If the final score is less than or equal to zero, then there is virtually no risk. If the final score is between one and ten, then the risk is low. If the final score is between eleven and twenty, then the risk is medium. If the final score is above twenty, then the risk is high.

Next, the risk level is compared to the benefit of the application, such as cost savings and revenue gained by the application, to determine whether to certify the application for deployment, as indicated by the following table:

Sav-ings	High Revenue	Medium Revenue	Low Revenue
High	No Risk = Certify	No Risk = Certify	No Risk = Certify
High	Low Risk = Certify	Low Risk = Certify	Low Risk = Certify
High	Medium Risk = *Certify	Medium Risk = *Certify	Medium Risk = Mitigate
High	High Risk = Mitigate	High Risk = Mitigate	High Risk = Mitigate
Med.	No Risk = Certify	No Risk = Certify	No Risk = Certify
Med.	Low Risk = Certify	Low Risk = Certify	Low Risk = Certify
Med.	Medium Risk =*Certify	Medium Risk = Mitigate	Medium Risk = Mitigate
Med.	High Risk = Mitigate	High Risk = Mitigate	High Risk = Reject
Low	No Risk = Certify	No Risk = Certify	No Risk = Certify
Low	Low Risk = Certify	Low Risk = Certify	Low Risk = Mitigate
Low	Medium Risk =*Certify	Medium Risk = Mitigate	Medium Risk = Reject
Low	High Risk = Mitigate	High Risk = Reject	High Risk = Reject

Note that "*Certify" means that this certification is contingent upon use of additional or more comprehensive risk mitigation processes such as implementation of intrusion detection, additional network segmentation or limitation of user access to application.

Based on the foregoing, a system, method and program product for determining whether to certify an application for deployment has been disclosed. However, numerous modifications and substitutions can be made without deviating from the scope of the present invention. For example, the values added and subtracted in the different processes can be modified, and additional or fewer factors can be considered in these processes. For example, the following, additional factors can be considered: current customer satisfaction in the customer sensitivity process, types of users supported in the customer sensitivity process, development standards followed for application in the mitigation process and complexity of application such as lines of code in the extent of vulnerability process.